

Replicate QDLinker with Google Colab

Wenbo Wang, New Jersey Institute of Technology

Agenda



- Background
- QDLinker introduction
- Related work
- Why I choose Google colab
- Steps to replicate

Background

- The growing needs to explore more and more spatial data and concepts.
- GIS related software have too many functions [1]
- There are more than 26,000 questions waiting to be answered in the [GIS Q&A forum](#).

[1] S. Gao and M. F. Goodchild, "Asking Spatial Questions to Identify GIS Functionality," 2013 Fourth International Conference on Computing for Geospatial Research and Application, San Jose, CA, 2013, pp. 106-110, doi: 10.1109/COMGEO.2013.18.

QDLinker Introduction

QDLinker

Learning to Answer Programming Questions with Software Documentation through Social Context Embedding

Jing Li^{a,*}, Aixin Sun^a, Zhenchang Xing^b

^a*School of Computer Science and Engineering, Nanyang Technological University, Singapore*

^b*College of Engineering and Computer Science, Australian National University, Australia*

Abstract

Official software documents provide comprehensive coverage but not aim for specific programming tasks or use cases. Often there is a mismatch between the documentation and questions encountered in specific programming tasks because of different wordings. We observe from Stack Overflow that **the best answers to programmers' questions often contain links to formal documentation.** In this paper, we propose a novel *deep learning to answer* framework, named **QDLinker**, for answering programming questions with software documentation. QDLinker learns from the large volume of discussions in **community-based question answering** to bridge the semantic gap between programmers' questions and software documentation. Specifically, QDLinker

li, Jing & Sun, Aixin & Xing, Zhenchang. (2018). Learning to answer programming questions with software documentation through social context embedding. *Information Sciences*. 448. 10.1016/j.ins.2018.03.014.

An example of dataset

How do I compare strings in Java?

▲ 727 I've been using the `==` operator in my program to compare all my strings so far. However, I ran into a bug, changed one of them into `.equals()` instead, and it fixed the bug.

▼ Is `==` bad? When should it and should it not be used? What's the difference?

★ 605 `java` `string` `equality` share edit flag edited Jan 23 '13 at 13:36 community wiki Nathan H

▲ `==` tests for reference equality (whether they are the same object).

3438 `.equals()` tests for value equality (whether they are logically "equal").

▼ `Objects.equals()` checks for nulls before calling `.equals()` so you don't have to (available as of JDK7, also available in [Guava](#)).

✓ Consequently, if you want to test whether two strings have the same value you will probably want to use `Objects.equals()`.



Query: How do I compare strings in Java?

Software documentation: `java.util.Objects.equals()`

<http://docs.oracle.com/javase/8/docs/api/java/util/Objects.html#equals-java.lang.Object-java.lang.Object->

Empirical study

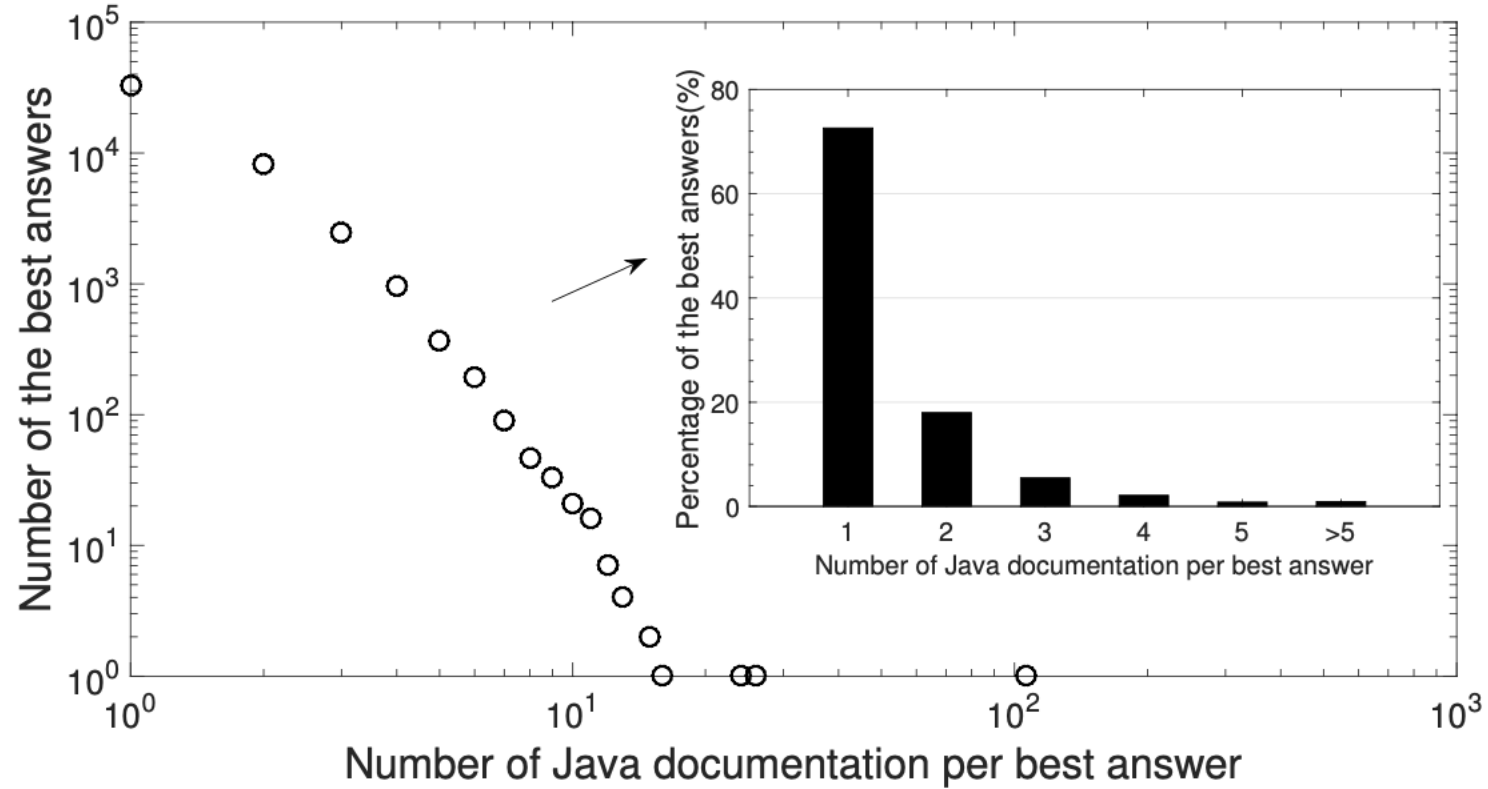
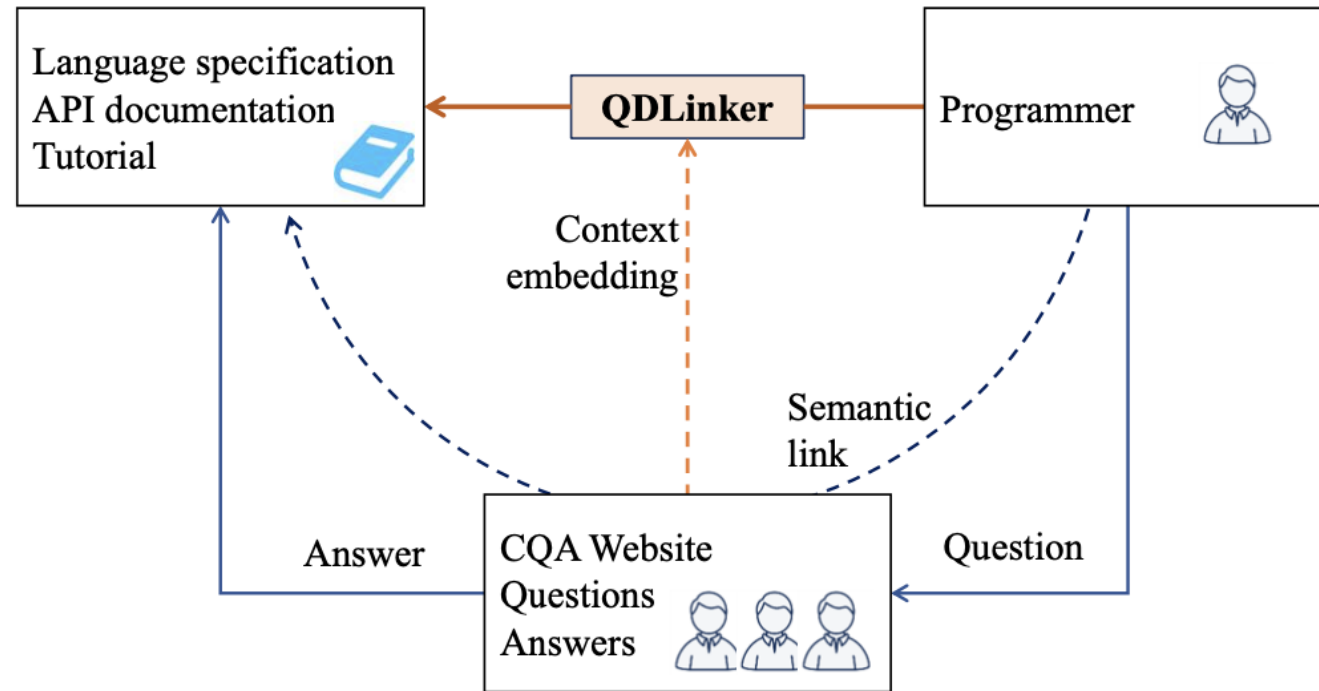
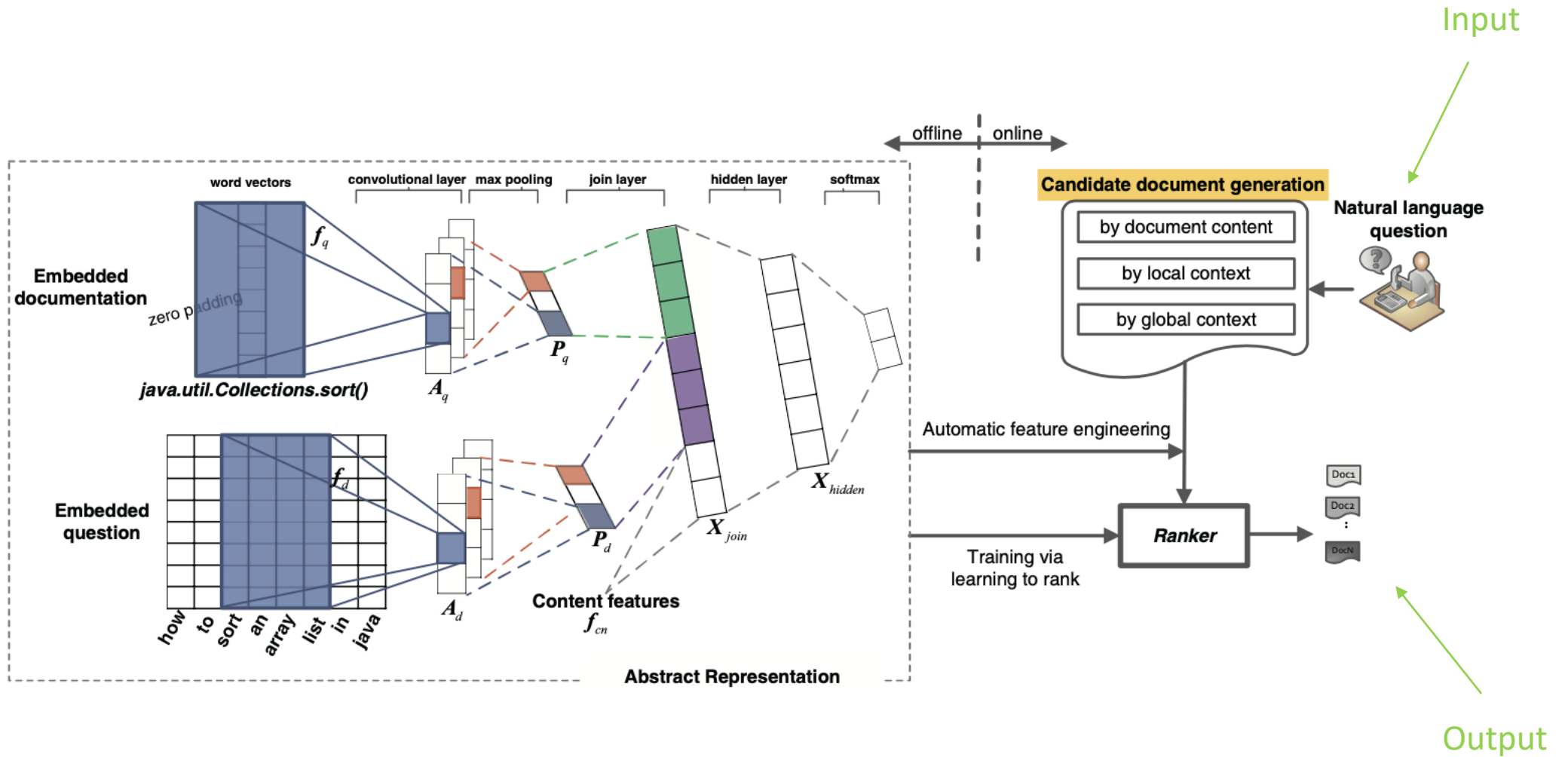


Figure 1: Distribution of links to Java documentation per best answer, among 45,288 best answers from Stack Overflow. The absolute numbers are plotted in **log scale**, and the percentages are plotted in bar chart.

Overview of QDLinker



Architecture of QDLinker



a) Social Context Embedding

- Two vocabularies
 - English words
 - Links to software documentation.
- Models
 - Mikolov's continuous bag-of-words and skip-gram languages (implemented in Word2vec)

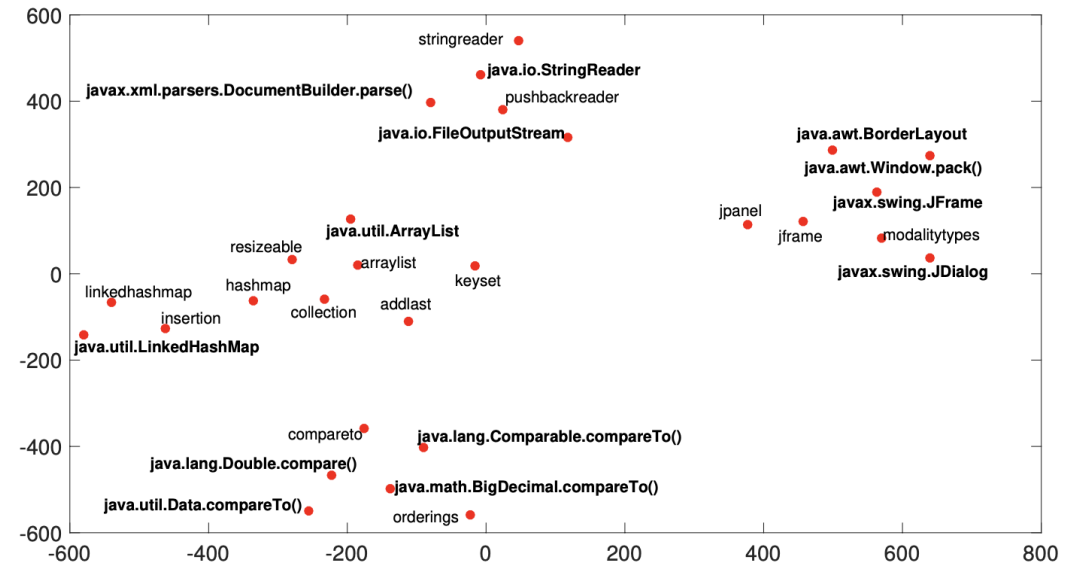


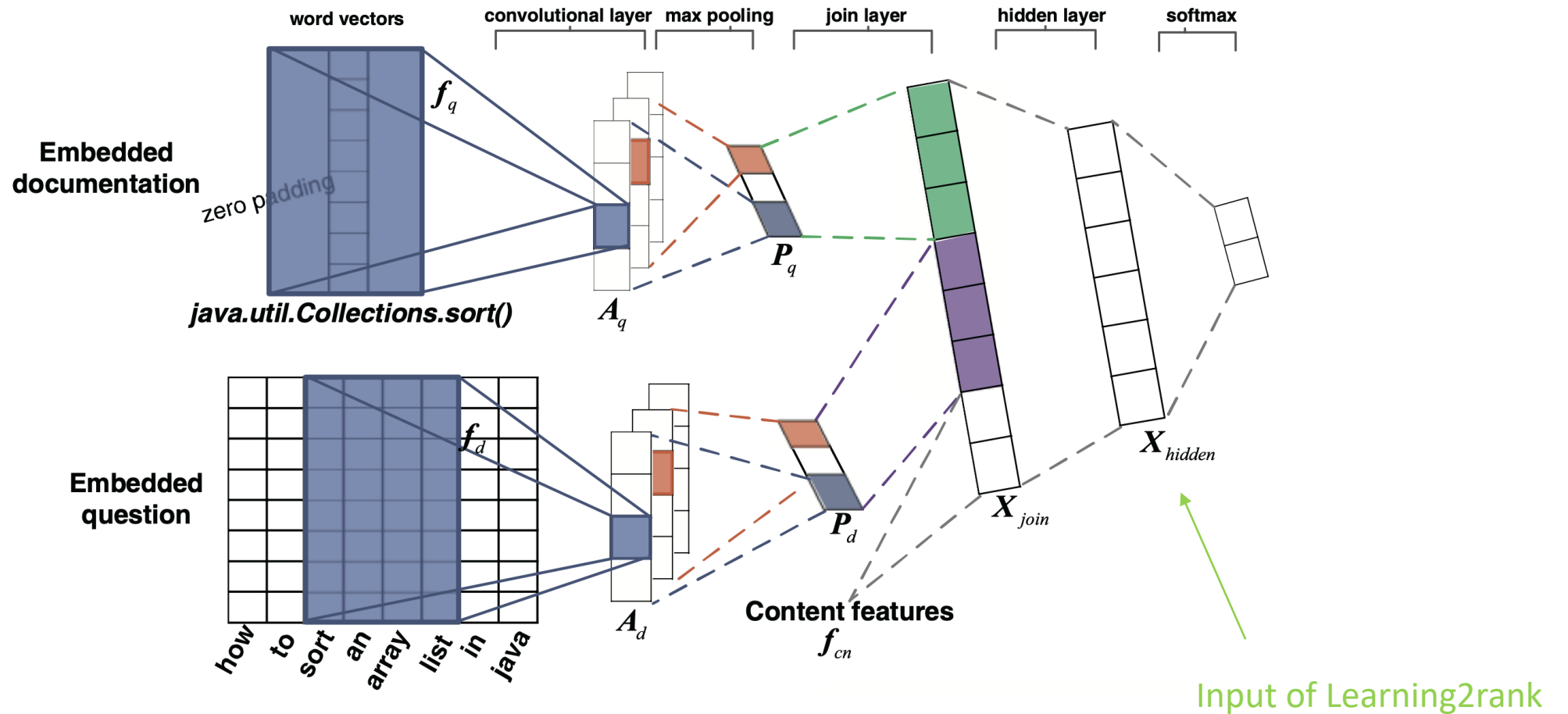
Figure 5: A 2D projection of embedding natural language terms and API documentation using PCA (API documentation in bold font and natural language terms in non-bold font).

Mikolov, Tomas & Sutskever, Ilya & Chen, Kai & Corrado, G.s & Dean, Jeffrey. (2013). Distributed Representations of Words and Phrases and their Compositionality. Advances in Neural Information Processing Systems. 26.

b) Candidate Document Generation

- **Document Content.** The content of software document reflects its relevance to a given query.
- **Local Context.** When a software document appears in a discussion thread, its surround texts reflect its relevance to the query question.
- **Global Context.** The global context of a software document is the collection of all its local contexts.

c) Four-layer Deep Neural Network



d) Learning a Ranker

- Learn a scoring function $F(q, d)$:

$$F(q, d) = \sum_{k=1}^K \omega_k \cdot \phi_k(q, d)$$

where each feature $\phi_i(q, d) \in \mathbf{X}_{hidden}$ measures a specific relationship between the query and a candidate software document. ω_i is the weight of the i -th feature (among the total K features), and is learned during the training process.

- Train the ranking model using LambdaMART

Related Work

Related work - 1

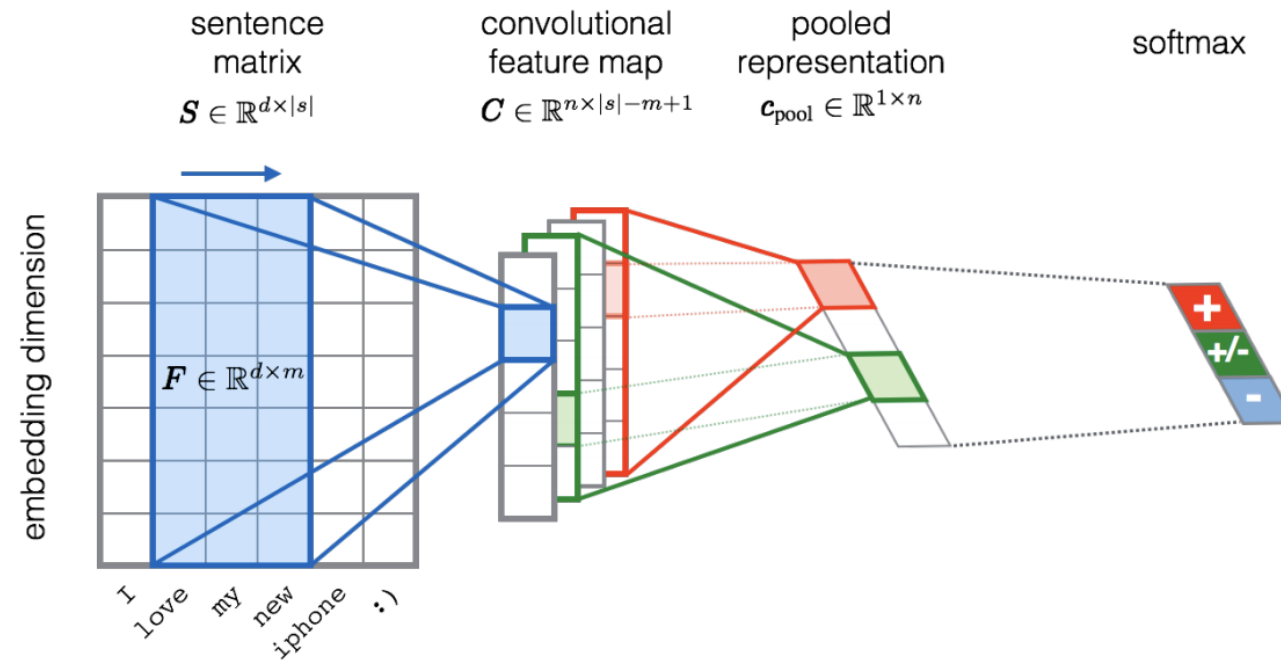


Figure 1: The architecture of our deep learning model for sentiment classification.

Severyn, Aliaksei & Moschitti, Alessandro. (2015). UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification. 10.18653/v1/S15-2079.

Related work - 2

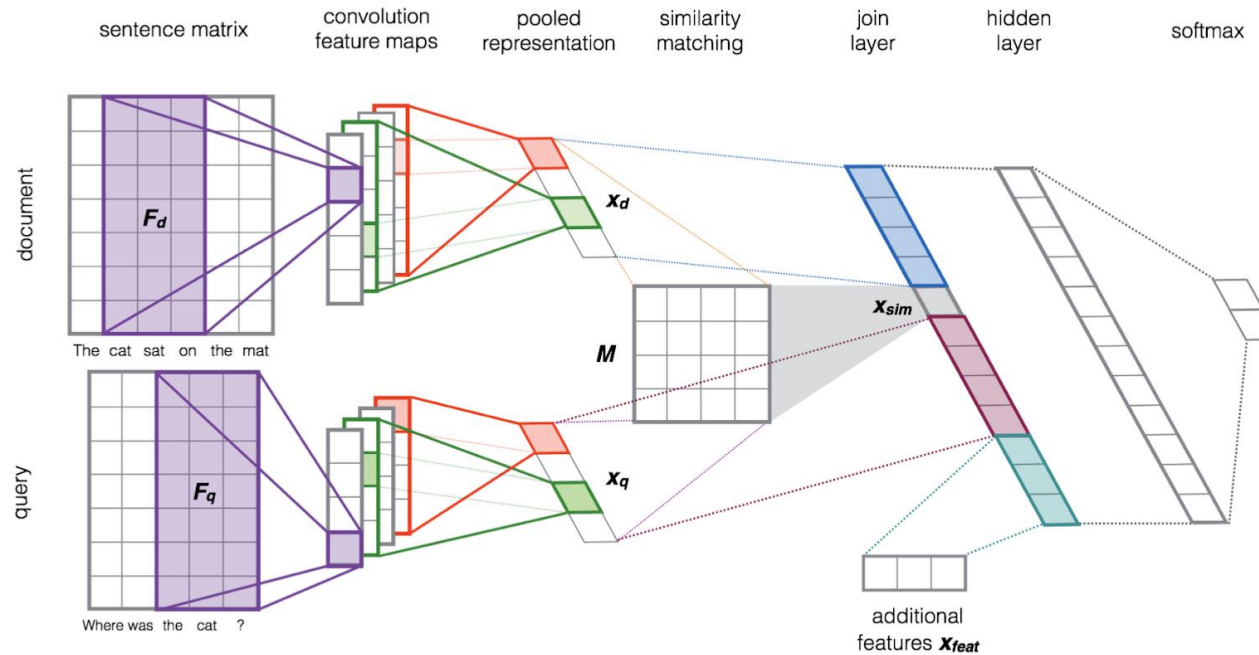


Figure 2: Our deep learning architecture for reranking short text pairs.

A. Severyn, A. Moschitti, Learning to rank short text pairs with convolutional deep neural networks, in: SIGIR, 2015, pp. 373–382.

Related work - 3

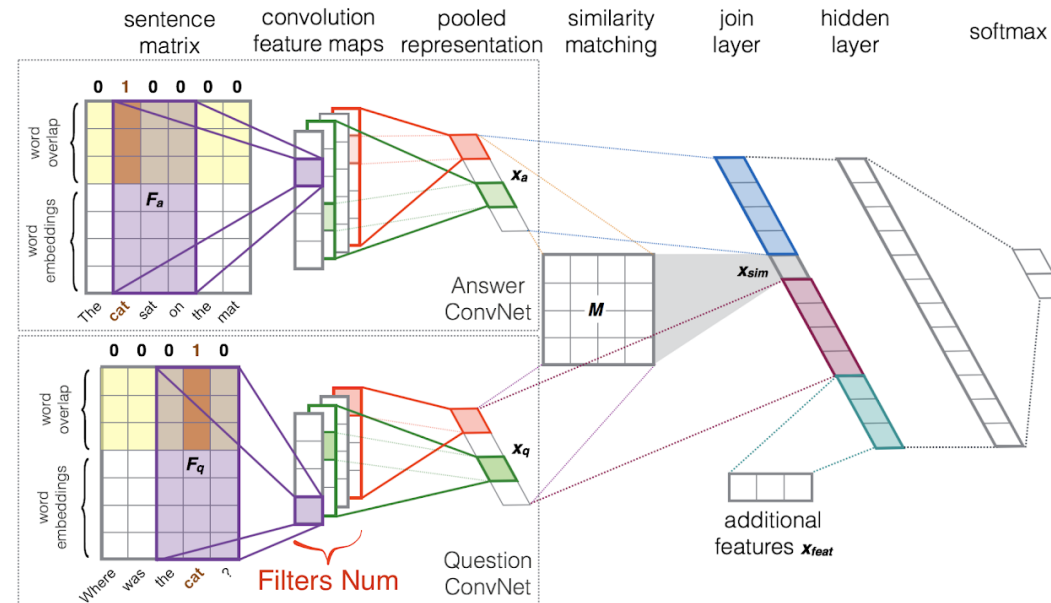
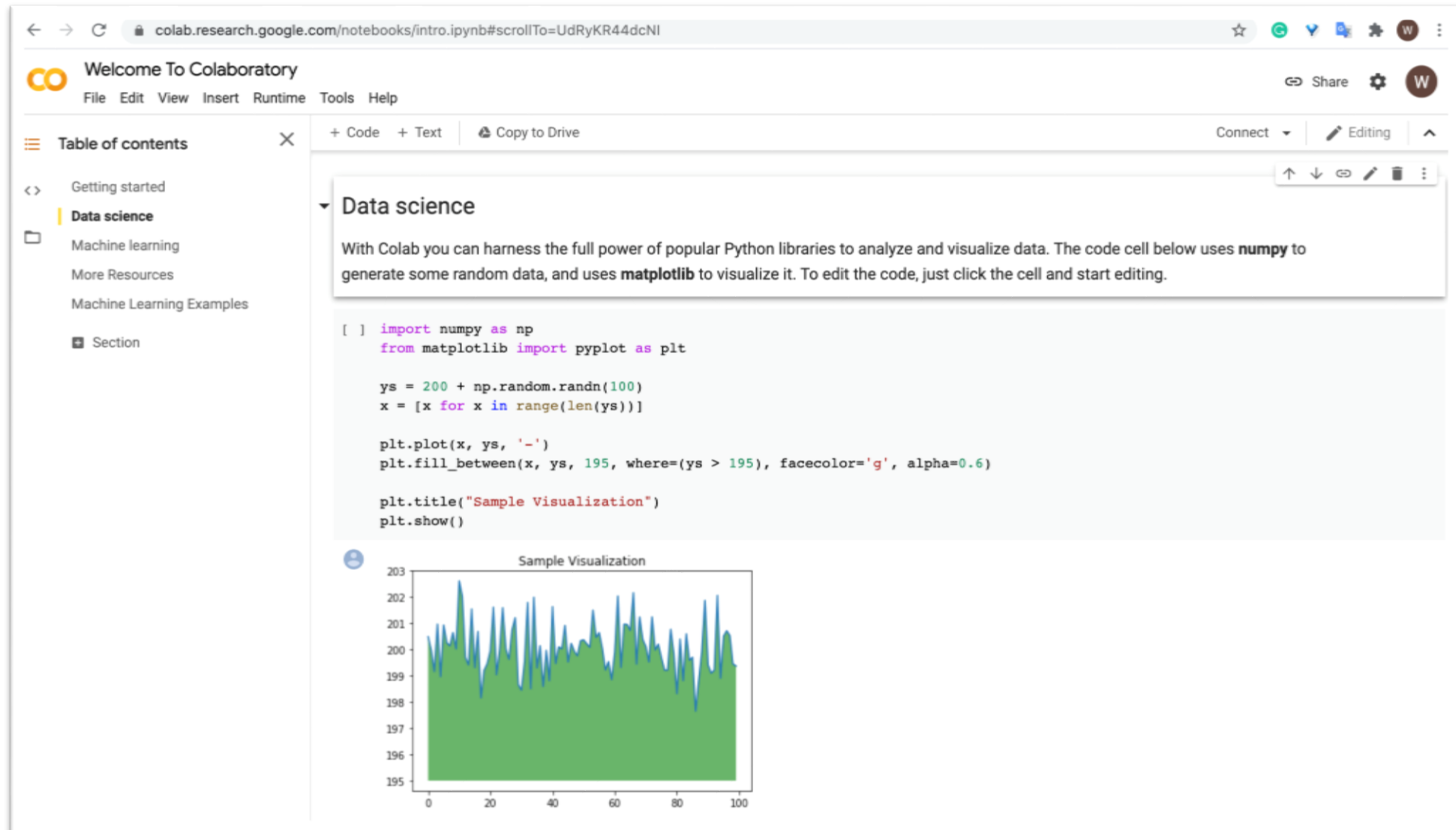


Figure 2: Our deep learning architecture for reranking question-answer pairs. The relational information in a pair is modelled by augmenting word embeddings with additional dimensions to encode **overlapping words**, e.g., we feed the network with additional *word overlap* indicator features whose values equal to 1 correspond to words that overlap in a pair, e.g., a non-stop word `cat`.

Aliaksei Severyn and Alessandro Moschitti. 2016. Modeling relational information in question-answer pairs with convolutional neural networks. arXiv preprint 1243 arXiv:1604.01178 (2016).

Why I choose Google colab

Google Colab



The screenshot displays the Google Colaboratory web interface. At the top, the browser address bar shows the URL `colab.research.google.com/notebooks/intro.ipynb#scrollTo=UdRyKR44dcNI`. The page header includes the Colab logo, the text "Welcome To Colaboratory", and a menu with options: File, Edit, View, Insert, Runtime, Tools, and Help. On the right side of the header, there are icons for "Share", "Settings", and a user profile icon.

On the left side, a "Table of contents" sidebar is visible, listing sections such as "Getting started", "Data science" (which is currently selected), "Machine learning", "More Resources", "Machine Learning Examples", and "Section".

The main workspace shows a code cell titled "Data science". Below the title is an introductory paragraph: "With Colab you can harness the full power of popular Python libraries to analyze and visualize data. The code cell below uses **numpy** to generate some random data, and uses **matplotlib** to visualize it. To edit the code, just click the cell and start editing."

The code cell contains the following Python code:

```
[ ] import numpy as np
from matplotlib import pyplot as plt

ys = 200 + np.random.randn(100)
x = [x for x in range(len(ys))]

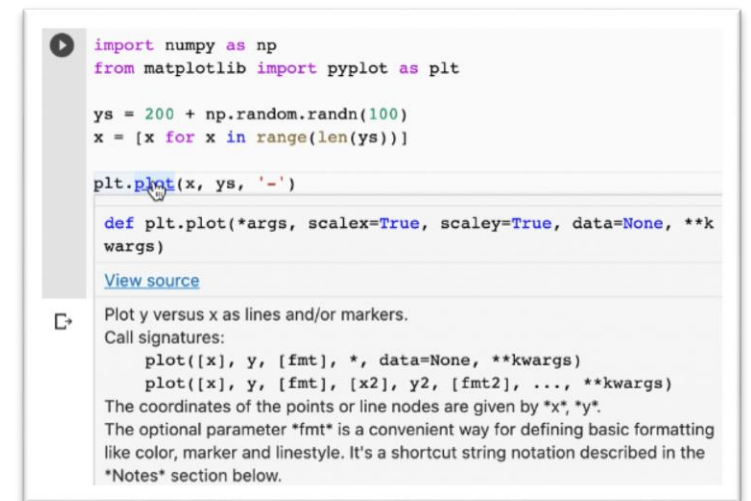
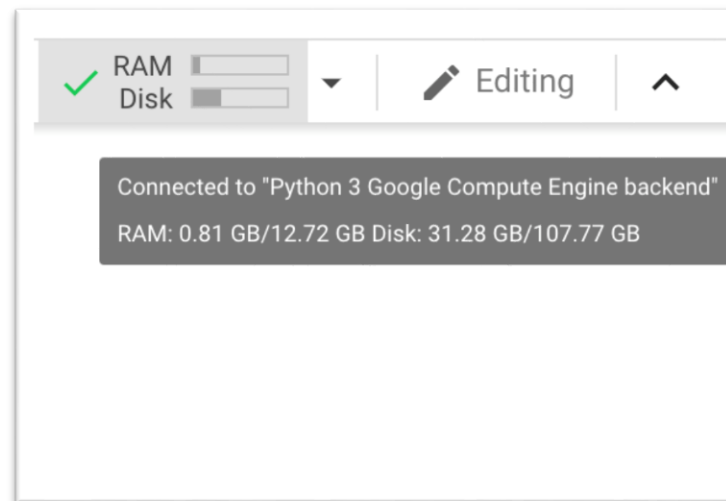
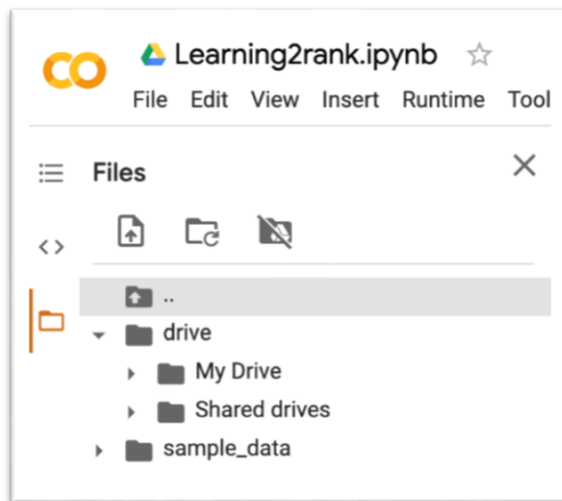
plt.plot(x, ys, '-')
plt.fill_between(x, ys, 195, where=(ys > 195), facecolor='g', alpha=0.6)

plt.title("Sample Visualization")
plt.show()
```

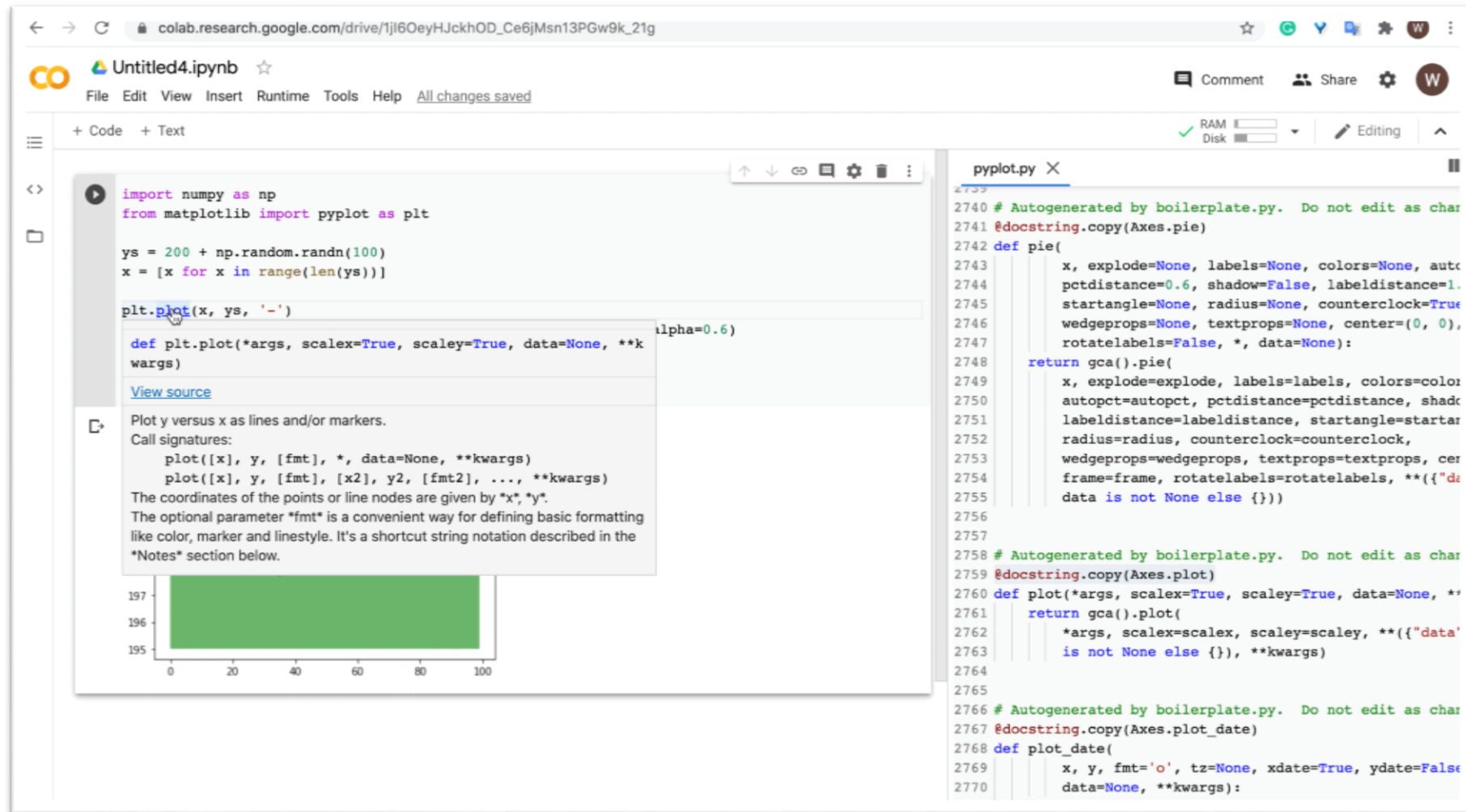
Below the code cell, a plot titled "Sample Visualization" is displayed. The plot shows a line graph with a blue line representing the data points and a green shaded area representing the region where the data points are greater than 195. The x-axis ranges from 0 to 100, and the y-axis ranges from 195 to 203.

Why I choose Google Colab

- Connect with Google Drive
- With GPU
- Syntax tips, highlighting
- Share with others
- Easy to debug



Examples



The screenshot shows a Google Colab notebook interface. The browser address bar displays `colab.research.google.com/drive/1j16OeyHJckhOD_Ce6jMsn13PGw9k_21g`. The notebook title is "Untitled4.ipynb". The code cell contains the following Python code:

```
import numpy as np
from matplotlib import pyplot as plt

ys = 200 + np.random.randn(100)
x = [x for x in range(len(ys))]

plt.plot(x, ys, '-')

def plt.plot(*args, scalex=True, scaley=True, data=None, **kwargs):
    alpha=0.6
```

A tooltip is visible over the `plt.plot` function call, providing documentation for the function. The tooltip text is:

Plot y versus x as lines and/or markers.
Call signatures:
`plot([x], y, [fmt], *, data=None, **kwargs)`
`plot([x], y, [fmt], [x2], y2, [fmt2], ..., **kwargs)`
The coordinates of the points or line nodes are given by *x*, *y*.
The optional parameter *fmt* is a convenient way for defining basic formatting like color, marker and linestyle. It's a shortcut string notation described in the *Notes* section below.

The plot shows a green line with markers representing the data points. The x-axis ranges from 0 to 100, and the y-axis ranges from 195 to 197.

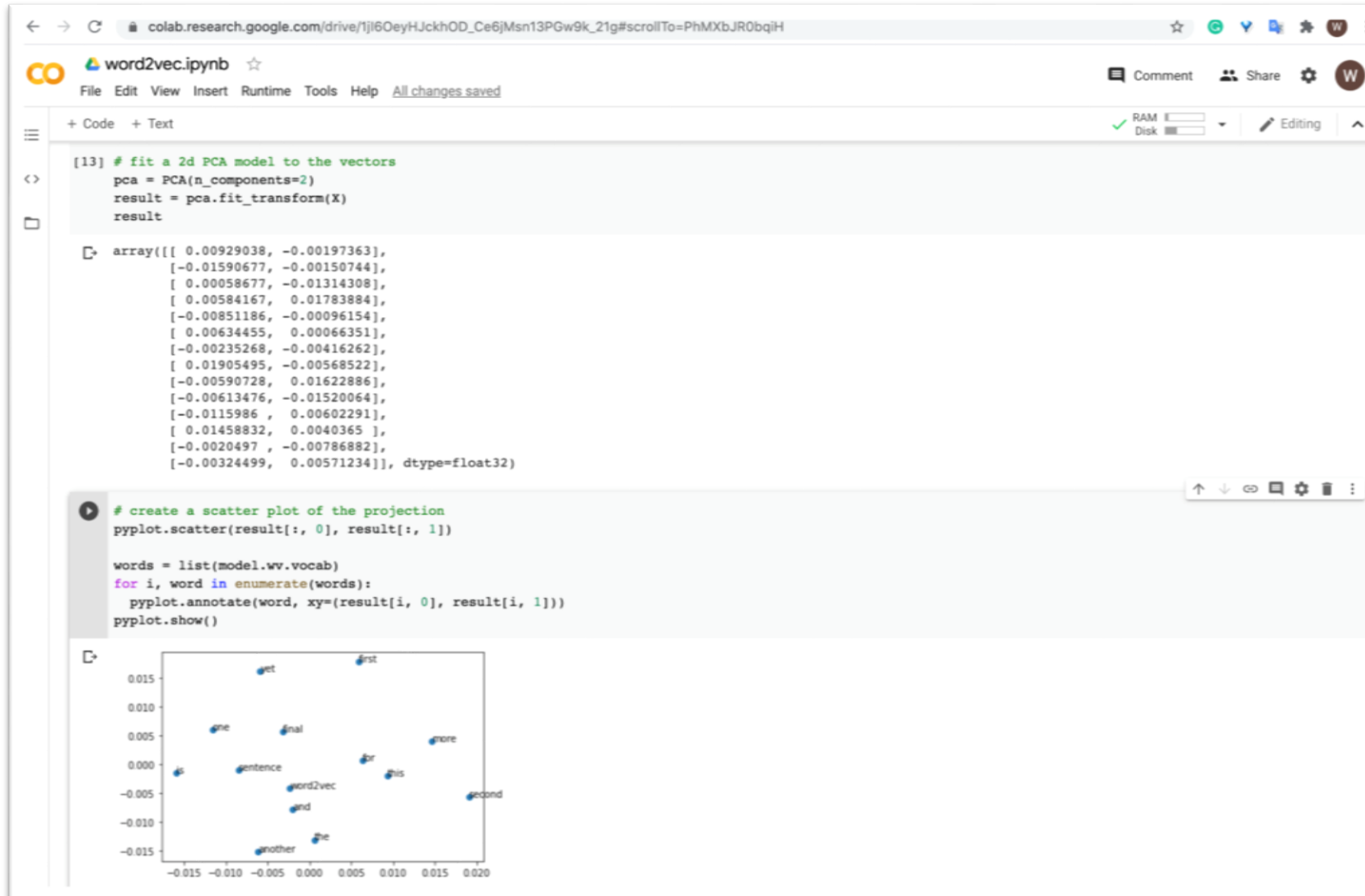
The right side of the notebook shows the source code for the `plt.plot` function, which is autogenerated by boilerplate.py. The code includes docstrings and function definitions for `plt.plot` and `plt.plot_date`.

Steps to Replicate

Steps to replicate

1. Read and understand the paper
2. Find existing solutions/tools/code
3. Test code and data (with Google Colab)
4. Complete the final code (with local develop environment)

Word2vec



The screenshot shows a Google Colab notebook titled "word2vec.ipynb". The code cell contains the following Python code:

```
[13] # fit a 2d PCA model to the vectors
pca = PCA(n_components=2)
result = pca.fit_transform(X)
result
```

The output of the code is a 2D array of PCA results:

```
array([[ 0.00929038, -0.00197363],
       [-0.01590677, -0.00150744],
       [ 0.00058677, -0.01314308],
       [ 0.00584167,  0.01783884],
       [-0.00851186, -0.00096154],
       [ 0.00634455,  0.00066351],
       [-0.00235268, -0.00416262],
       [ 0.01905495, -0.00568522],
       [-0.00590728,  0.01622886],
       [-0.00613476, -0.01520064],
       [-0.0115986 ,  0.00602291],
       [ 0.01458832,  0.0040365 ],
       [-0.0020497 , -0.00786882],
       [-0.00324499,  0.00571234]], dtype=float32)
```

Below the array, there is another code cell:

```
# create a scatter plot of the projection
pyplot.scatter(result[:, 0], result[:, 1])

words = list(model.wv.vocab)
for i, word in enumerate(words):
    pyplot.annotate(word, xy=(result[i, 0], result[i, 1]))
pyplot.show()
```

The output of this code is a scatter plot showing the projection of words onto a 2D space. The x-axis ranges from -0.015 to 0.020, and the y-axis ranges from -0.015 to 0.015. The plot shows several words plotted as points, with their corresponding labels:

- et
- rst
- ne
- nal
- more
- sentence
- word2vec
- his
- end
- another
- he
- second

Learning to rank

The screenshot shows a Google Colab notebook interface. The browser address bar displays `colab.research.google.com/drive/1wwwsl-hQdQJTUpMI4nwEft3w3uj1YBDq#scrollTo=zdoCdzVnpkwn`. The notebook title is "Learning2rank.ipynb". The left sidebar shows a file explorer with a folder named "sample_data". The main area contains a code cell with a table of training metrics for iterations 50 to 850. The metrics include values for C, B, and S, along with a time column. The training process ends with "Early termination at iteration 877" and a time of "0:15:56.171242". Below the table is a code cell with the following Python code:

```
Epred = model.predict(EX)
print('Random ranking:', metric.calc_mean_random(Eqids, Ey))
print('Our model:', metric.calc_mean(Eqids, Ey, Epred))
```

The output of the code cell shows:

```
Random ranking: 0.27258472902087394
Our model: 0.47573893617277513
```

At the bottom left, a disk usage indicator shows "76.40 GB available".

Future work

- A Web interface
- A Chrome extension

Geographic Information Systems

Home
Questions
Tags
Users
Unanswered

Search Results

Advanced Search Tips [Ask Question](#)

Results for choropleth of calculated voronoi polygons

Choropleth of calculated Voronoi polygons? [Search](#)

1 results

Relevance Newest Votes Active

1	qgis_docs_print_composer	0.12421
2	qgis_docs_geometry_tools	0.01601
3	qgis_docs_geometry	0.01601
4	qgis_docs_clipping_and_merging_raster_layers	0.01601
5	qgis_docs_modifying_vector_layers	0.01601

Choropleth of calculated Voronoi polygons?

... a voronoi layer for a set of points, I then joined said layer with counts so (almost) each point polygon has a number associated to it. My question is: how do ... I color each polygon in QGIS? My data is simply the voronoi layer (I don't exactly know where it gets the data polygons), an ID for each point that generated ...

... onoi-thiessen

asked Nov 29 '16 by [Dervin Thunk](#)

Code

- [Word2vec\(colab\)](#)
- [Learning2rank\(colab\)](#)
- [DNN\(colab\)](#)
- <https://github.com/jma127/pyltr>
- <https://sourceforge.net/p/lemur/code/HEAD/tree/RankLib/>

Reference

- li, Jing & Sun, Aixin & Xing, Zhenchang. (2018). Learning to answer programming questions with software documentation through social context embedding. *Information Sciences*. 448. 10.1016/j.ins.2018.03.014.
- Mikolov, Tomas & Sutskever, Ilya & Chen, Kai & Corrado, G.s & Dean, Jeffrey. (2013). Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems*. 26.
- Q. Wu, C. J. Burges, K. M. Svore, J. Gao, Adapting boosting for information retrieval measures, *Information Retrieval* 13 (3) (2010) 254–270.
- S. Gao and M. F. Goodchild, "Asking Spatial Questions to Identify GIS Functionality," 2013 Fourth International Conference on Computing for Geospatial Research and Application, San Jose, CA, 2013, pp. 106-110, doi: 10.1109/COMGEO.2013.18.
- Severyn, Aliaksei & Moschitti, Alessandro. (2015). UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification. 10.18653/v1/S15-2079.
- A. Severyn, A. Moschitti, Learning to rank short text pairs with convolutional deep neural networks, in: *SIGIR*, 2015, pp. 373–382.
- Aliaksei Severyn and Alessandro Moschitti. 2016. Modeling relational information in question-answer pairs with convolutional neural networks. arXiv preprint 1243 arXiv:1604.01178 (2016).